The quantum cluster (QCT) code : User's manual

David Sénéchal

version August 7, 2006

1 General principles

The `qct` code applies quantum cluster methods to the Hubbard model and performs various types of calculations, such as

1. Ordinary CPT spectral functions for energy distribution curves (EDCs) and momentum distribution curves (MDCs).
2. The VCPT grand potential Ω and the search for saddle points.
3. CDMFT calculations
4. The order parameters associated with the various terms in the Hamiltonian.
5. Systematic loops over variational parameters in calculating Ω or the CDMFT distance function.

It is written in `C++` and borrows many utility routines (integration, etc.) from other sources. The command-line is `qct` and requires options, as well as input files.

Input files required The code requires a parameter file, called `para.dat` by default. The format of this file is described below. It also requires a cluster description file, usually with the suffix `.clus`, whose precise name is specified in the parameter file.

2 Command-line options

If the command `qct` is entered without any option, a list of available options, with a short description, is printed to the screen. The same is true if the option `-h` is typed.

`-aaph`

Calculated the self-energy functional Ω (SEF) with the method explained in the paper by Aichhorn, Arrigoni, Potthoff and Hanke (cond-mat/0607271): the frequency integral is calculated analytically (formula by Potthoff) and the integrand of the wavevector integral is evaluated by the diagonalization of a full $M \times M$ matrix, M being the number of iterations in the band Lanczos method. This method is actually considerably slower than the one used by default in this code: the band Lanczos method followed by a numerical integration over frequencies. It is provided for sake of comparison only.

`-cdmft`

Uses the CDMFT self-consistency loop instead of a minimization of the grand potential. Uses a fixed grid for frequencies and wavevector. Reads the variational parameters like for the option `-nr` below. The self-consistency loop stops when the relative change in the distance function is less than `CDMFT_REL_ACCUR` (defined in `cdmft.C`).

`-cf`

Uses the continued fraction method to calculate the cluster Green function. This was the only method in used before August 2006, but is much slower than the band Lanczos method which is now the default. It is however recommended when the dimension of the Hilbert space is small (say less than ten times the number of orbitals).

`-cg`

Optimizes the SEF with the conjugate gradient method. Since the SEF is not a minimum in general by a saddle point and the conjugate gradient method only finds minima (or maxima), this options uses in alternance distinct conjugate gradient procedures for the variables for which the SEF is a minimum (i.e. the Weiss fields) and for the chemical potential, for which it is a maximum. This option is slower than `-nr`, but more robust, and can be used as a first stage of search. It stops when the position of the variational parameters changes by less than `accur_cg`, set in the parameters file.

`-cluster_data`

Writes to a file the cluster ground state expectation value of the number of electrons and spin at each sites, as well as the corresponding correlations.

`-dos`

Calculates the density of states and puts it in 5-column format in the file ⟨*output*⟩`_dos`. Range of frequencies specified as in option `-s` below. The five columns are: (1) frequencies, (2) lattice DOS for spin up (normalized to one), (3) lattice DOS for spin down (normalized to one), (4) cluster DOS for spin up (normalized to one), (5) cluster DOS for spin down (normalized to one).

`-full`

In constructing the hamiltonian, builds a full sparse matrix for H instead of using tensor factors for the off-diagonal terms. Not useful in practice (takes more memory and slower, contrary to initial expectation).

`-grad`

Calculates the gradient of the SEF at the end of a CDMFT run.

`-ksym`

Symmetrizes the Green function over a point group. This option needs an argument: R, or D.

- R means rotational symmetry by $\pi/2$ and exchange $x \leftrightarrow y$, so that 8 terms are combined.
- D means diagonal symmetry $x \leftrightarrow y$ only (two terms are combined).

`-loop`

Does a loop over variational parameters and writes the grand potential (or the distance function) to a file. The variational parameters are specified in the parameter file between the keyword `LOOP_PARAMETERS` and the next asterisk (`*`). Each variational parameter is specified on a separate line, with (i) its name, (ii) the starting value, (iii) the end value and (iv) the increment. A maximum of 3 nested loops is allowed. By default, it calculates the grand potential Ω for each value and appends a line in the file ⟨*output*⟩`_sef`. Otherwise, if the option `-cdmft` is specified, it is the corresponding distance function that is calculated.

`-mdc`

Calculates the CPT spectral function as a momentum distribution curve (MDC). The following values are read in the parameter file:

- `freq` : the real part of the frequency.
- `k_eta` : the imaginary part of the frequency (for broadening).
- `k_step` : the step in wavevector

Depending on the `ksym` option, the range of wavevectors is either the top half of the Brillouin zone (case `D`) or the first quadrant (case `R`).

Example:

```
freq 0
k_eta 0.05
k_step 0.02
```

`-mvcpt`

Applies the VCPT procedure (Newton-Raphson or Quasi-Newton) to an increasingly large body of parameters. Variational parameters are either of the symmetry-breaking type (S) or of the normal type (N). Examples of N-type parameters are the chemical potential or a hopping term. Examples of S-type parameters are the AF Weiss field M of the various dSC Weiss fields. This option first reads the list of variational parameters (see the `vcpt` option). Then, in a first procedure, only treats the type N parameters as variational, setting the other ones to zero; in other words, it looks for the normal state solution. Then it considers each S-type parameter in turn, varying it together with all the N-type parameters. If it fails to find a non trivial solution, it tries again with a larger starting value (or smaller, if the procedure led to a divergent value). At last, it varies all parameters together, using the previous converged values as starting points. This is the best option to use if one wants a more complete picture of the variational space.

`-nloop`

Performs a loop over the chemical potential μ and another parameter (typically U) so as to find and then stay at the conditions for a preset filling. Syntax:

`target_n` $\langle n \rangle$ $\langle \Delta n \rangle$
`n_loop` $\langle param \rangle$ $\langle start \rangle$ $\langle end \rangle$ $\langle increment \rangle$ $\langle \Delta \mu \rangle$

where n is the target density (e.g. $n = 1$ for half-filling), Δn is the required precision for n (e.g. 0.001), $\langle param \rangle$ is the name of the parameter to be varied (e.g. U), $\langle start \rangle$,$\langle end \rangle$,$\langle increment \rangle$ are the starting, end and increment values of the parameter, and $\Delta \mu$ is the step in μ to be taken to progress towards half-filling. The advantage of this method (over the `-nmin` option below) is that the parameters evolve in a smooth way, thus allowing the solutions to be found in a continuous way from the previous point's solution.

This option must be specified in conjunction with either `cdmft`, `nr`, `cg` or `prop`, to specify the type of action taking place at each value of the parameter. The converged values of the variational parameters found for each value of the loop parameter are used as initial values for the next run (that is the whole point of this option).

`-nmin`

Performs a super-loop over chemical potential in order to achieve a preset value of density n. Does a `-o` at each value of `mu` and uses the `zbrent` routine to find the preset value of n. The following values are read in the parameter file:

- `target_n` : target density, followed par required accuracy (see option `-n_loop` above).
- `mu1` : the lower bound for μ.
- `mu2` : the upper bound for μ.

`-nr`

Looks for an extremum point of Ω in the space of variational parameters. Uses the Newton-Ralphson method. The initial point must be set with common sense. The variational parameters are specified in the parameter

file between the keyword `VARIATIONAL_PARAMETERS` and the next asterisk (`*`). Each variational parameter is specified on a separate line, with (i) its name, (ii) the starting value, (iii) the minimum value and (iv) maximum value the parameter can take (acceptable range) and (v) the initial incremental step the parameter should be given in the search. *Example:*

```
VARIATIONAL_PARAMETERS
mu        0.5        -5        5          0.05
tx        0.25       -3        3          0.02
*
```

The procedure stops when the calculated gradient is inferior to the parameter `accur_grad` set in the parameters file.

Each time the grand potential is evaluated in the optimization procedure, the values of the parameters and grand potential is appended to ⟨*output*⟩`_sef`. The value of the extremum found, its properties, as well as the values of the first and second derivatives for each variable, are written to a file whose name contains the names of the variational parameters, prefixed by `var_`.

At the end of the procedure, the properties of the extremum are calculated (i.e. the `-prop` option is run), except when the procedure fails to converge to a value within bounds. See also `mvcpt`.

`-ploop`

Does a loop over a lattice parameter (e.g. chemical potential or U). Syntax:

`p_loop` ⟨*param*⟩ `loop` ⟨*start*⟩ ⟨*end*⟩ ⟨*increment*⟩

or

`p_loop` ⟨*param*⟩ `list` ⟨N⟩ ⟨*list of N numbers*⟩

Examples:

`p_loop mu loop 2.4 3.2 0.05`

or `p_loop mu list 7 2.4 2.5 2.6 2.7 2.8 2.9 3.0`

This option must be specified in conjunction with either `sc`, `vcpt`, `sc_vcpt` or `prop`, to specify the type of action taking place at each value of the parameter. The converged values of the variational parameters found for each value of the loop parameter are used as initial values for the next run, unless the option `reinit` is specified.

`-prop`

Calculates the order parameters associated with each term of the Hamiltonian (or "properties"). Basically, for a diagonal, hopping or anomalous term in the cluster Hamiltonian, the expectation value of the corresponding operator is calculated. If the hopping or anomalous term is defined with a link type (see below), then the corresponding expectation value takes into account inter-cluster hopping or pairing. More specifically, for a normal one-body term defined as

$$h \sum_{i,j,\sigma} s_{ij} \left(c^\dagger_{i,\sigma} c_{j,\sigma} + H.c. \right)$$

the corresponding order parameter is defined as

$$-\frac{2}{L} \Im \int_C \frac{\mathrm{d}\omega}{2\pi} \sum_{\mathbf{K}} \mathrm{tr}(\hat{s}(\mathbf{K}) \hat{G}_{\mathrm{cpt}}(\mathbf{K}, \omega))$$

where $\hat{s}$ is the matrix s_{ij} and depends on $\mathbf{K}$ (the reduced wavevector) is it corresponds to a hopping term with inter-cluster hopping. The traces includes a sum over spins. In the case of anomalous terms, a similar form is used (in practice, the Nambu formalism is used so there is no big difference between a hopping term and an anomalous term). The frequency integral is taken along a contour C that half-circles all (we hope) negative-axis poles of the Green function.

The output is appended to a file called `prop.dat`. Two lines are written each time: a commented one for the description of parameters (with column numbers) and the following for their values. The first four columns give: (1) the name of the cluster description file; (2) the value of U; (3) the value of the density of electrons on the cluster; (4) the value of the grand potential. The remaining columns give the values of the various parameters defined in the cluster description file, as well as their expectation value. If inter-cluster values are different from their default, they are also given.

`-qn`

In VCPT, uses the quasi-Newton method instead of the Newton-Raphson method (see option `-rn` above). Takes less time when a large number of variational parameters is used.

`-reinit`

Reinitializes the parameters each time when doing a loop over a lattice parameter (`-ploop`, instead of starting from the variational parameters found for the previous value of the parameter.

`-s`

Computes the CPT spectral function (EDCs) and places the output in the file ⟨*output*⟩`_sp`. The first column is for freqencies, the others for the various wavevectors sweep specified by `wavevectors` in the parameter file. The wavevectors are written on the first (commented) line of the file. The following values are read in the parameter file:

- `wavevectors` : the wavevector sweep (see section below)
- `eta` : the imaginary part of the frequency
- `wmin` : the lower bound of the plot
- `wmax` : the upper bound of the plot
- `step` : the frequency step
- `period_option` : the type of periodization used (0 by default)

`-seed`

Sets the seed for the random number generator used in setting the initial state in the Lanczos algorithm. Needs the seed as an argument. Useful to check whether a strange result can be attributed to a degeneracy of the ground state.

`-sef`

Calculates the SEF (i.e. the grand potential Ω) for the set of parameters specified in the parameter file. Appends the result in ⟨*output*⟩`_sef`.

`-self_dist`

When doing CDMFT, uses a distance function weighed by the norm of the self-energy $\Sigma(\omega)$, instead of a flat (constant) weight.

`-sym`

Uses the cluster symmetries to shorten the calculation. The cluster symmetries need to be specified in the cluster description file. In addition, the symmetries are checked against the particular parameters used and only those that are really present in the matrix `t` of the code are really used (this is much safer than in previous versions). This option can save a lot of time on larger clusters and is heavily recommended.

`-test`

Used for debugging. Action may depend on version and stage of development. Not for general use.

3 The cluster description file

The cluster description file is specified following the keyword `cluster_file` in the parameter file. The best way to understand the structure of that file is to look at an example.

Here is a description of the generic parameters specified in that file:

- `dim` : the spatial dimension of the model (1, 2 or 3).
- `L` : the number of sites L of the cluster.
- `nb` : the number of bath sites (zero if omitted).
- `nband` : the number of bands in the model (set to unity if omitted).
- The optional keyword `no_Nambu`, which means that the Nambu formalism will not be used, and that the model can involve the operators S_x and S_y can be defined, but without anomalous terms.
- `sites` : the (x, y) positions of the sites, as integers, preceded by the number (label) of each site (from 1 to L). For `dim` > 1 only.
- `neighbors` : the basis vectors of the superlattice of clusters (again in (x, y) format). For `dim` > 1 only.

Following these generic parameters, various one-body terms in the Hamiltonian are specified. These are of three types: (i) interaction terms, (ii) link-based operators and (iii) local operators.

Interaction terms If `nband`=1, the interaction is taken as the usual Hubbard interaction U for each site on the cluster (bath excluded). Nothing to write.

For multi-band models, the interactions must be specified explicitly, for instance, by:

```
INTERACTION
U     i     1     1     (0)   1
Up    i     2     2     (0)   1
V     i     1     2     (0)   1
*
```

each line describes an interaction parameter, in this order: (1) its name, (2) the keyword `i` (for 'interaction'), (3) the two band indices concerned, (4) the displacement vector (always zero for an interaction) and (5) the multiplier (usually unity).

Link-based operators Link-based operators are hopping terms (normal or anomalous) that are normally defined for the whole lattice but are restricted to the cluster. If an operator (like NN hopping) is defined using links, its average value computed with the lattice Green function will take into account the inter-cluster link as well. Lattice hopping terms must be defined like this.

A typical definition of link operators in the cluster description file would be:

```
LINK_OPERATORS
tx    h     b_1   b_2   ΔR    ⟨multiplier⟩
Dx    s     b_1   b_2   ΔR    ⟨multiplier⟩
px    t     b_1   b_2   ΔR    ⟨multiplier⟩
*
```

where the columns are :

- The name of the parameter.
- the type of link:
  - `h` : hopping term
  - `s` : singlet superconducting pairing term
  - `t` : triplet superconducting pairing term
  - `x` : off-diagonal spin pairing term (if `no_Nambu` is set).
- The band indices (omitted if `nband` = 1).
- The link direction, e.g., $\Delta\mathbf{R}$ = `(1,0)` for a NN link in the x direction in 2D.
- A multiplier, e.g., -1 for all hopping terms, by convention. The multiplier can be opposite for singlet pairing operators in the x and y directions, for instance, so that a d-wave pairing is obtained if those two parameters are equal.

Local operators Local operators are defined by specifying the explicit terms in the hopping matrix, without explicit reference to an inter-cluster extension. The list of operators must be contained between the keyword `OPERATORS` and an asterisk (`*`). Here is an example:

```
OPERATORS
tb1u          4          h          n
1 +           1 +        -1
2 +           2 +        -1
3 +           3 +        -1
4 +           4 +        -1

eb1u          4          b          n
1 +           1 +        1
2 +           2 +        1
3 +           3 +        1
4 +           4 +        1

d1            8          b          s
1 +           2 -        -1
3 +           4 -        -1
1 +           3 -        1
2 +           4 -        1
5 +           6 -        -1
7 +           8 -        -1
5 +           7 -        1
6 +           8 -        1

M             8          c          s
1 +           1 +        1
1 -           1 -        -1
2 +           2 +        -1
2 -           2 -        1
3 +           3 +        -1
3 -           3 -        1
4 +           4 +        1
4 -           4 -        -1
*
```

The first line of each operator descriptions specifies (1) the name of the parameter, (2) the number of lines in the list of terms that follows, (3) whether the operator is based on the cluster (`c`), on the bath (`b`) or is a hybridization term between the two (`h`) and (4) whether the parameter is of N-type (`n`) or of S-type (`s`) [see the `mvcpt` option above for explanations]. Then a number of lines follows for each operator giving : (1) the label of the first site with its spin (+ or −) (2) the label of the second site with its spin (+ or −) the value of a multiplier.

Important notes:

- Hermitian conjugates are taken care of automatically and must not be specified in order to avoid double counting.
- site labels are counted from 1 to L for cluster sites and from 1 to n_b for bath sites. In the hybridization case (h), cluster sites are specified in the first column, bath sites in the second column.
- A space must be kept between the site label and the spin label (i.e. `1 +` and not `1+`).

4 The parameter file

Each parameter defined in the cluster description file has a *cluster* value and a *lattice* value. The values of the lattice parameters are read in the parameter file, between the keyword `LATTICE_PARAMETERS` and the next asterisk (`*`). Each parameter is specified on its own line, by its name followed by its numerical value.

Example:

```
LATTICE_PARAMETERS
U 2
mu 1.5
tx 0.25
t1 0.075
t2 0
*
```

If a parameter is not specified, its value is assumed to be zero,*unless* this parameter has a default parameter, in which case its value is taken to be the same as that of the default parameter. If a parameter is specified that does not exist in the model, a simple warning is issued on the screen, but the program is otherwise unaffected.

The values of the cluster parameters is by default the same as that of the lattice parameters specified. The default values are overridden by the list specified between the keyword `CLUSTER_PARAMETERS` and the next asterisk (`*`). Some parameters, like the Weiss fields, should not normally have a nonzero lattice value and are necessarily specified here.

Example:

```
CLUSTER_PARAMETERS
mu 1.1
tx 0.3
M 0.14521
Dx 0.3721
*
```

Variational and loop parameters are specified the like with the keywords `CLUSTER_PARAMETERS` and `LOOP_PARAMETERS` (See the corresponding options in the first section of this document).